Barunastra ITS for RobotX 2026: Nala Atlas, Icarus, and Jovan

D. A. Briet, M. M. Miserideityo, J. N. Dikairono, D. V. Aubin, T. R. Nareswari, Y. K. Hendrata, R. M. Ansori, W. S. Nugroho, F. R. Akbar, W. N. Irawan, A. J. E. Caraka, K. N. Pratama, G. Y. Prawira, S. S. Rasendria, D. M. Suryono, B. H. Yudanto, M. R. Kurniawan, R. A. Maulana, I. S. Nugraha, R. I. Rasyad, N. P. Koroy, R. H. Putri, Z. N. A. Syamsudin, G. N. Anggalaksmana, P. N. A. Demetrias, S. R. Jatnika, A. Talenta, V. Ardiansyah, M. A. Rafif, A. R. Nugraha, M. K. E. Jausyan, J. V. Margaretha, M. A. Ilham, A. Steventio, H. Jovan, F. A. Herali, M. A. Hidayat, K. Arnasati, J. C. Senodiputro, E. R. D. Raswanto, D. P. S. Arum, F. A. Meyori, R. Dikairono, L. Hakim, Muhtadin, M. I. Zazuli
Barunastra ITS, Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
barunastra.its@gmail.com

***Abstract*—This report presents the system design and technical strategy of Barunastra ITS for the 2026 Maritime RobotX Challenge. The system integrates an Unmanned Surface Vehicle (USV), Unmanned Underwater Vehicle (UUV), and Unmanned Aerial Vehicle (UAV) into a coordinated multi-platform System of Systems (SoS), with each platform performing complementary roles in navigation, underwater inspection, aerial observation, and mission execution. A ROS 2-based architecture enables communication and coordination between the independently developed platforms. Mission execution is designed around parallel operation and cross-platform information exchange, enabling platforms to coordinate their actions while maintaining independent onboard control. System reliability is evaluated through progressive testing from simulation to integrated multi-platform field trials, ensuring readiness for reliable and coordinated execution of the RobotX 2026 missions.**

Acronyms and Abbreviations

GCS	Ground Control Station
ROS 2	Robot Operating System 2
SoS	System of Systems
UAV	Unmanned Aerial Vehicle
USV	Unmanned Surface Vehicle
UUV	Unmanned Underwater Vehicle

I. About Team

Barunastra ITS is a multidisciplinary undergraduate robotics team from Institut Teknologi Sepuluh Nopember (ITS) with members from various academic majors. The team focuses on the development of Unmanned Surface Vehicle (USV) and has gained experience through research, engineering projects, and participation in the RoboBoat competition, including the 2026 edition. For the 2026 Maritime RobotX Challenge, Barunastra ITS works together with two other ITS robotics teams: Bayucaraka ITS, which develops the The Unmanned Aerial Vehicle (UAV) platform, and Banyubramanta ITS, which develops the Unmanned Underwater Vehicle (UUV) platform. Each team brings its own technical expertise, development experience, and knowledge gained from previous competitions and research activities. These capabilities are combined to develop the USV, UUV, and UAV as a coordinated multi-vehicle system, with the teams retaining responsibility for their respective platforms while collaborating on the interfaces, communication, and mission coordination required for RobotX 2026.

II. Competition Strategy

A. General Strategy

For this competition, our primary strategy is to maximize mission efficiency through parallel multi-platform operation. Rather than executing missions sequentially, the USV and UUV are deployed simultaneously whenever their operational areas and requirements allow. UAV

maintains a priority-based task queue containing mission requests from the other platforms, ensuring that critical missions are addressed first without unnecessarily interrupting ongoing operations.

This strategy allows each vehicle to operate according to its capabilities while maintaining a coordinated mission sequence. The USV provides surface navigation and LiDAR-based environmental information, the UUV performs underwater inspection and repair tasks, and the UAV provides aerial observation and data collection.

B. Mission Approach

The mission approach relies on a centralized ground communication layer, with Robot Operating System 2 (ROS 2) serving as the primary interface between all vehicles and RoboCommand. The USV, UUV, and UAV communicate directly with the Ground Control Station (GCS), which manages and routes mission-related data across the fleet.

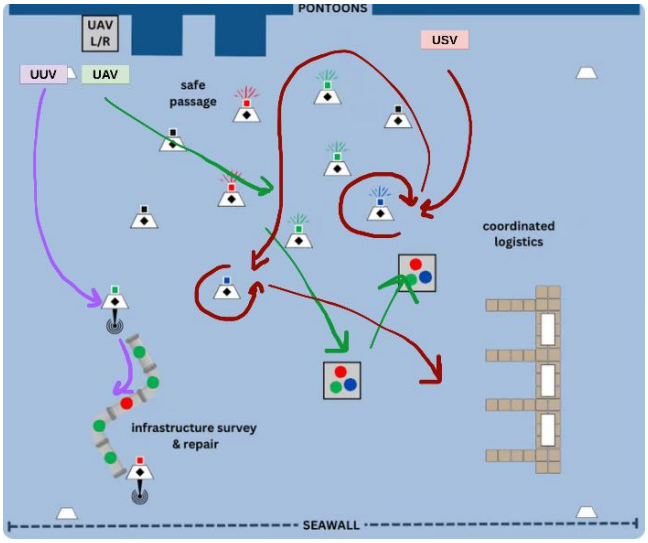


Fig. 1. Competition arena overview [1] with planned vessel trajectories: UUV (purple), UAV (green), and USV (red).

During mission initiation and execution, each platform acknowledges the status of other platforms before starting or transitioning between tasks. This acknowledgment mechanism ensures that mission states remain synchronized and prevents conflicting operations. The system prioritizes parallel execution whenever the missions can safely operate independently as shown on Fig. 1.

Task 1 - Safe Passage is executed through coordinated operation between the USV and UAV. The USV manages surface navigation through the passage, while the UAV provides supplemental aerial observation. The UAV fuses its onboard LiDAR, camera, and GPS data to identify objects and obstacles within the task area, communicating this mapping telemetry to the USV.

The USV generates a navigation plan based on the detected targets and the expected mission sequence. To maximize safety, the USV additionally employs a reactive local obstacle avoidance layer using its own LiDAR measurements to navigate immediate environmental constraints.

The system also accounts for disruptive changes in the environment. If a target's state shifts, the USV acknowledges the updated telemetry received from the UAV. Once the new state is confirmed, the USV dynamically replans its trajectory using the latest environmental data, preserving mission progress and minimizing unnecessary travel.

While the USV and UAV execute Task 1, the surfaced UUV simultaneously transits toward the **Task 2 - Infrastructure Survey and Repair** area by tracking the acoustic pinger. During transit, the UUV incorporates environmental data shared by the UAV and USV to improve navigation safety, using UAV-based object observations and the USV's LiDAR-generated map as supplementary navigation references.

Upon reaching the designated area, the UUV submerges and uses its onboard computer vision system to locate the pipeline. YOLO-based detection and object segmentation identify the pipeline and visual markers, providing the necessary geometric data for alignment and tracking. The UUV then follows the pipeline via visual servoing, continuously adjusting its surge, sway, and yaw based on the detected position and orientation.

When a red light is detected, the UUV transitions to the repair state and aligns its magnetic payload with the designated target. The platform performs station keeping while lowering

the payload and waits for the indicator to transition to green. Once the green indication is confirmed, the UUV records the subsequent color sequence for reporting.

After gathering the required data, the UUV resurfaces and transmits the color sequence telemetry to the UAV and GCS. The UAV logs this data into its dynamic task queue, allowing it to complete any ongoing missions before initiating the next phase. Once ready, the UAV surveys the area from an elevated altitude. Based on the sequence received from the UUV, the UAV then locate and retrieve the correct colored tin using its magnetic payload mechanism and successfully deliver it to the correct resource circle.

Upon completing Task 1, the USV immediately transits toward **Task 3 - Coordinated Logistics** area and searches for an appropriate docking location. The USV uses computer vision to survey the docking area and identify an available bay based on its indicator light. Once a suitable bay displaying the required green indication is identified, the USV uses the detected position to align and center itself before proceeding with the docking operation.

Once positioned inside the selected bay, the USV identifies the specific target using its visual indicator. Upon detecting a red light, the USV activates its water shooter, directing a sustained stream into the designated target window. The USV maintains this continuous stream until the visual indicator transitions to green.

Following the successful water delivery sequence, the USV records its newly displayed color sequence and transmits this data to the UAV and RoboCommand. Because the overall system operates asynchronously, the execution timelines of Task 2 and Task 3 typically overlap. Consequently, the UAV queues for this incoming telemetry from the USV without overriding its ongoing flight state. The UAV then repeats the payload operation described in Task 2, processing the USV's unique color sequence to retrieve the next designated tin and deliver it to its corresponding target location.

Task 4 - Dynamic Incident Response requires coordinated operation between the USV, UUV, and UAV. Upon receiving mission data and commands from RoboCommand via the centralized ground layer, the system assigns response duties based on the tier. During the Core Tier, the UAV is responsible for resolving aerial tasks, while the USV manages surface events. The UUV is not assigned as a primary response platform for this tier due to its inherent navigation limitations and operational constraints.

For the Advanced and Disruptive Tiers, each platform maintains independent planning and response logic. The planning system integrates directives received from RoboCommand with real-time observations shared across the fleet.

This distributed planning approach allows the system to respond to dynamic changes while ensuring safe navigation. Each platform leverages cross-platform telemetry to enhance situational awareness, while the centralized ground layer ensures that mission commands and operational states remain completely synchronized with RoboCommand.

III. Design Strategy

A. Platform Architecture

The physical architecture of our platforms follows a modular design approach, allowing each component to be inspected, replaced, or repaired independently. This makes maintenance easier and more efficient while reducing downtime during operations. Safety is also considered throughout the design process to ensure that each platform remains reliable and secure. The design approach for each platform is described below.

1) *Icarus (UAV):* The UAV is developed following a system-of-systems design philosophy, where modular hardware and software tools can be reconfigured to support complex missions. Each new tool is evaluated based on five criteria: (1) safety of personnel, vehicle, and environment; (2) modularity for rapid integration; (3) incremental development from a base-level capability; (4) performance-focused metrics for each subsystem; and (5) balancing novelty with anticipated performance and safety. For this competition, Icarus focuses on flight-control refinement, robust computer vision through calibration, synchronization, and training, full UAV autonomy, and reliable inter-vehicle communication through ROS 2 topics. These improvements support autonomous aerial observation, data collection, and coordination with the USV and UUV.

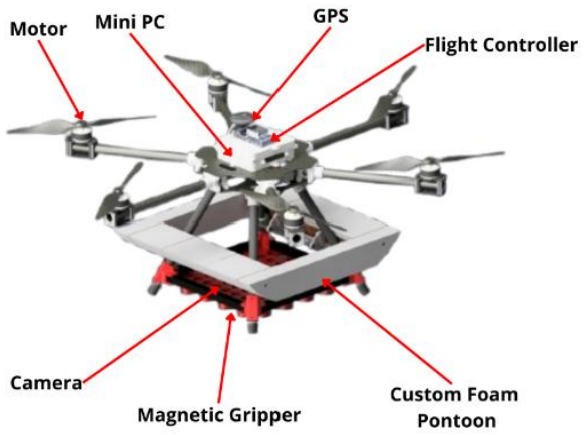


Fig. 2. Perspective view of Icarus.

2) *Nala Atlas (USV):* The physical architecture of the USV is based on a catamaran hull with an aluminum extrusion frame. Building on the successful performance of Ares in Roboboat 2026, Atlas adopts the same Aluminum Alloy 6063 extrusion framework to provide structural stiffness and efficient load distribution while minimizing weight. The extrusion-based design also enables modular hardware integration through sliding-nut assemblies, allowing rapid reconfiguration without permanent structural modifications.

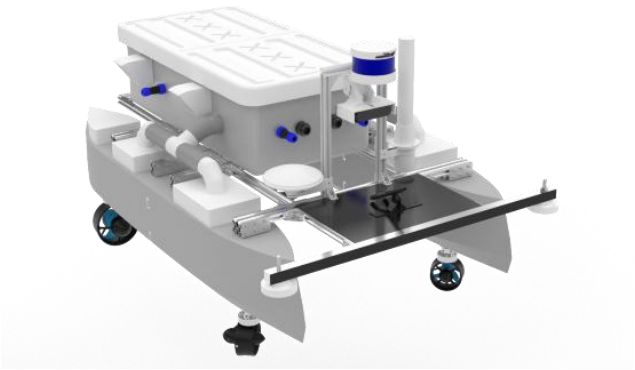


Fig. 3. Perspective view of Nala Atlas.

3) *Jovan (UUV):* The UUV uses a lightweight aluminum watertight enclosure with an internal screw-hole matrix, allowing flexible system configuration and future component upgrades. Pressure simulations in Autodesk Fusion were used to design the enclosure to withstand depths of up to 10 m. A cylindrical acrylic tube attached to the main enclosure houses the batteries, while both enclosures are supported by durable and lightweight HDPE frames.



Fig. 4. Perspective view of Jovan.

B. Propulsion

The propulsion systems are designed to balance maneuverability and transit efficiency, allowing each platform to adapt to different mission requirements. The USV utilizes four thrusters with two movement configurations: holonomic and azimuth, with holonomic movement providing greater stability and precision during operations, while the azimuth configuration is used for efficient transit between areas. The UAV utilizes six motors to provide sufficient thrust and precise movement for its total weight of approximately 5.8 kg. Meanwhile, the UUV is equipped with eight T-200 thrusters, providing high maneuverability in confined spaces and enabling six degrees of freedom movement through its distributed thruster configuration.

C. Mission Modules

The vessel's hardware is outfitted with specialized modules to execute mission-specific tasks. These components facilitate the transition from sensor-based detection to physical actuation as detailed below:

1) *Magnetic gripper*: A module consisting of 24 neodymium magnets arranged on the underside of the UAV, used for the tin retrieval mission. The wide magnetic contact area helps compensate for minor alignment errors and wind drift, ensuring the payload remains securely attached even when slightly off-center.

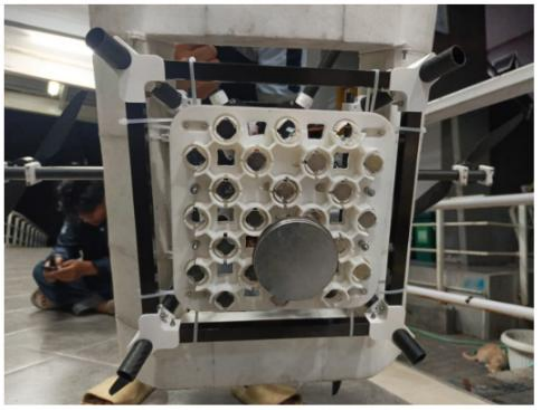


Fig. 5. Magnetic gripper module.

2) *Water shooter module:* A water-shooting system equipped with two underwater pumps, used for the water-shooting mission. An inlet diameter of 0.86 cm is designed to maximize fluid acceleration, achieving a pressure of 180 kPa and a velocity of 19 m/s to maintain a stable jet against wind resistance.

3) *Magnetic plate:* A magnetic plate mounted underneath the UUV, used for the object retrieval mission. A waterproof camera is positioned behind the plate to provide visibility during descent and assist with alignment and object detection.

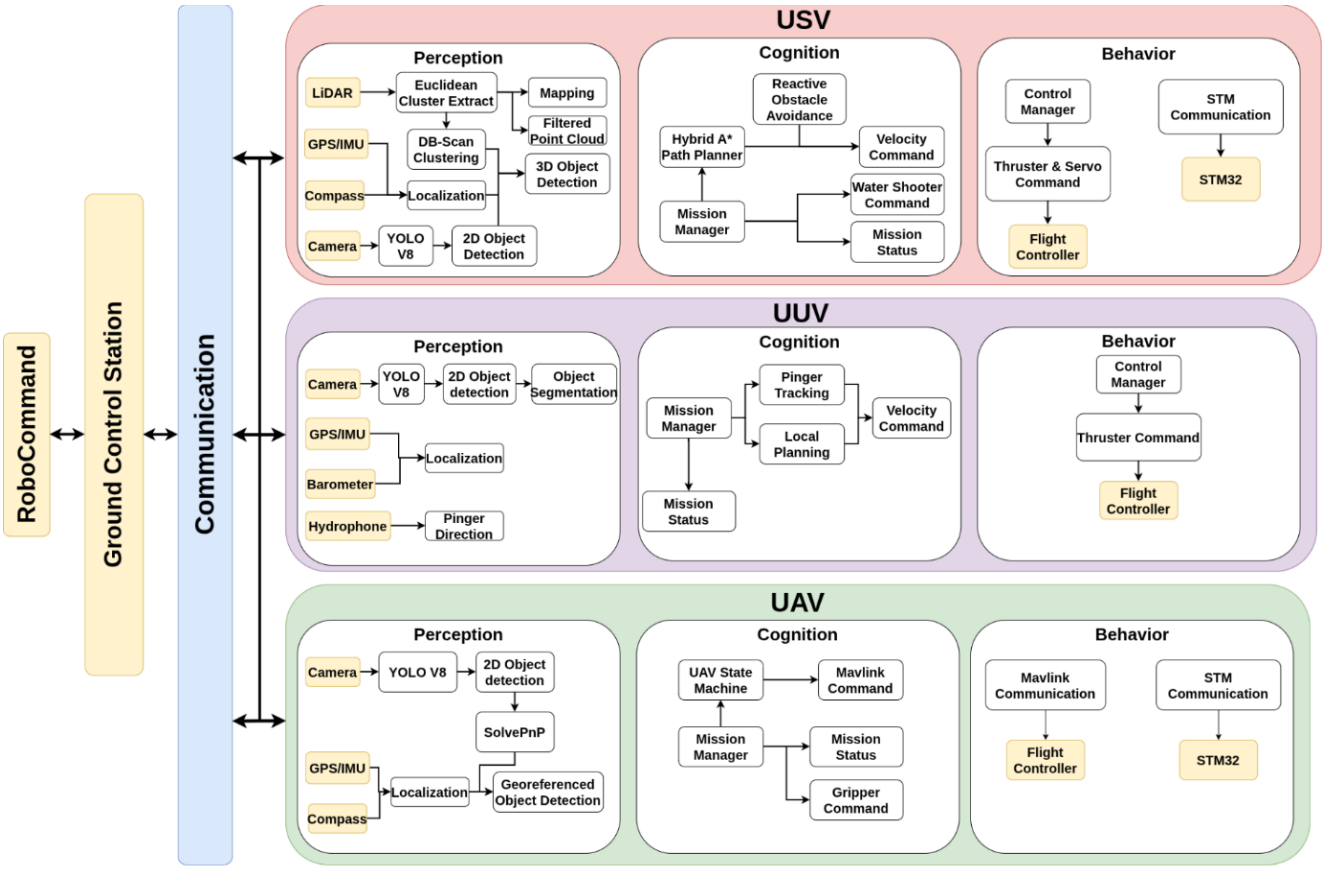


Fig. 6. Software architecture diagram.

D. Software Architecture

The USV, UUV, and UAV were initially developed independently by three distinct teams, resulting in varying software structures and implementation approaches. To integrate these platforms into a cohesive System of Systems (SoS), a common software architecture was established. This architecture provides a shared structure, terminology, and separation of responsibilities, enabling independently developed components to interoperate through well-defined interfaces.

The software architecture is divided into four main layers: Perception, Cognition, Behavior, and Communication.

1) ***Perception module****:* Acquires and interprets sensor data to determine the platform's state and surrounding environment. This output feeds into the Cognition layer and, when necessary, is shared with other platforms.

a) *UAV:* Relies primarily on a downward-facing camera for aerial observation. A YOLO-based 2D object detection model identifies mission-relevant targets, converting image coordinates into georeferenced positions using camera calibration, platform attitude, altitude telemetry, and OpenCV's solvePnP pose estimation. The solvePnP method estimates camera/object pose from corresponding 3D and 2D points using the camera's intrinsic parameters [2].

b) *USV:* Fuses LiDAR, camera, GPS, IMU, and compass data to construct a comprehensive environmental representation. LiDAR point clouds are processed using Euclidean Cluster Extraction to identify and filter environmental structures, while YOLOv8 handles 2D visual detection. These inputs are calibrated and merged with localization data to estimate object positions in three-dimensional space.

c) *UUV:* Leverages visual, acoustic, inertial, and depth measurements for underwater perception. The camera pipeline employs YOLO to detect relevant structures, followed by segmentation to estimate object geometries and centerlines. An acoustic locator provides directional bearings toward the pinger, which are integrated with IMU and depth telemetry to continuously estimate the platform's state.

2) ***Cognition module****:* Interprets perception data, manages mission execution, and determines the platform's subsequent actions.

a) *UAV:* Utilizes a control node containing a mission state machine to manage operational phases, including takeoff, climb, survey, hover, transit, approach, visual servoing, return, and landing. Task-specific logic dictates transitions between GPS-based navigation, area surveys, and vision-based precision positioning. During Tasks 2 and 3, detected targets are used to center the platform over the designated delivery area.

b) *USV:* Coordinates the overall mission sequence and generates task-specific commands via a central Mission Manager. Navigation goals are routed to a Hybrid A* planner for feasible global path generation, while a reactive obstacle avoidance module provides local responses to dynamic obstacles and environmental disturbances. The implementation uses the Nav2 Smac Hybrid-A* planner, which provides a cost-aware Hybrid-A* search with motion models such as Dubins and Reeds-Shepp [3].

c) *UUV:* Employs an asynchronous Behavior Tree to coordinate mission states and respond to Perception layer inputs. This framework organizes task execution hierarchically through nodes that return states such as SUCCESS, FAILURE, and RUNNING; asynchronous actions remain in the RUNNING state while extended operations execute [4]. This structure manages transitions between pinger-based transit, pipeline following, repair operations, and resurfacing.

Mission decisions can be based not only on locally generated perception data but also on information received from other vehicles through the Communication layer.

3) ***Behavior module****:* Translates commands generated by Cognition into physical platform actions. It provides the interface between high-level mission decisions and hardware-specific actuation, including thrusters, motors, servos, flight controllers, and mission payloads.

a) *UAV:* Converts navigation and payload decisions into flight-control and actuator commands. Position-hold and survey commands govern general navigation, while visual servoing refines the platform's position during final approach. It interfaces with the PX4 flight controller via external/offboard control, where externally generated position, velocity, attitude, or other setpoints are provided to the flight stack [5].

b) *USV:* Utilizes a Control Manager to translate generated velocity commands into thruster and servo actuation according to the active motion mode. The platform supports azimuth, differential, and holonomic control modes, allowing its actuation strategy to adapt to varying mission requirements. Propulsion and steering commands are transmitted to the flight controller, while auxiliary hardware and mission

payloads are managed through dedicated STM32 communication pipelines.

c) *UUV:* Executes navigation commands through closed-loop velocity control. Active depth control mitigates vertical drift during underwater operations, while a lease-based goal arbitration mechanism ensures that only one maneuver command drives the actuators at any given time. During mission-state transitions, a zero-velocity barrier is applied to eliminate residual platform momentum before a new maneuver is initiated.

4) ***Communication****:* Provides the interface among the individual platforms, the ground control system, and the overarching SoS. ROS 2 serves as the primary middleware for exchanging mission commands, platform status, perception results, and task-completion telemetry across distributed software components. This architecture enables each platform to operate independently while exposing relevant data to other systems when coordination is required.

At the system level, the Communication layer acts as the integration boundary between the independently developed platform software stacks. Rather than requiring the internal implementations of the USV, UUV, and UAV to be identical, only the data exchanged across this boundary must adhere to agreed interfaces and semantics. This paradigm allows the three platforms to retain their specialized onboard architectures while functioning cohesively as a coordinated multi-platform system.

IV. Testing Strategy

To maximize development efficiency and mitigate technical risk, the testing strategy follows a structured plan consisting of three progressive phases: simulation, off-water component validation, and on-water field trials. Software modules and control logic are validated in simulation while the physical hardware is being developed, allowing the core software to be tested before the platforms are deployed on water.

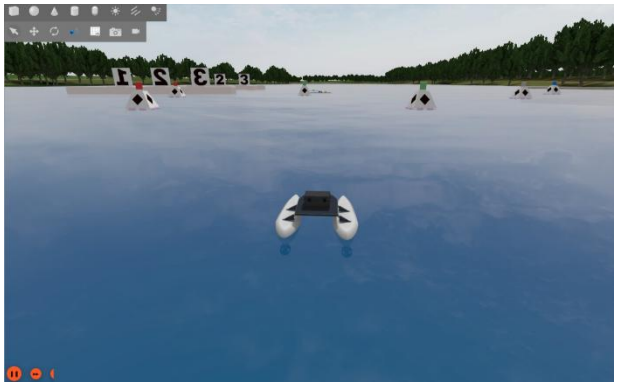


Fig. 7. Custom Gazebo simulation testing for USV.

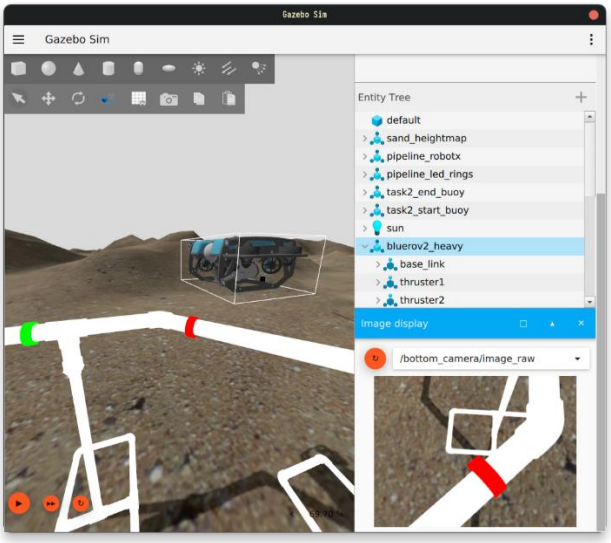


Fig. 8. Custom Gazebo simulation testing for UUV.

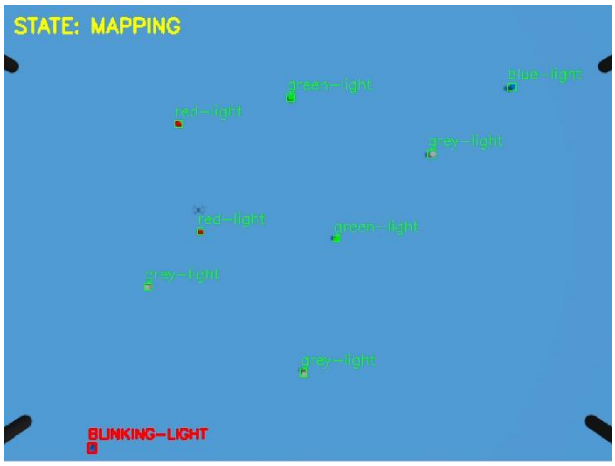


Fig. 9. Custom Gazebo simulation testing for UAV.

The software architecture is designed to transition from simulation to physical deployment, allowing the same modules to run in

Gazebo and onboard the physical platforms. Gazebo serves as the primary environment for evaluating navigation, control tuning, and computer vision. By recreating the RobotX 2026 tasks virtually, the Perception and Cognition layers can be evaluated in a controlled environment. Running the ROS 2 middleware in simulation also allows message exchange and multi-platform coordination issues to be identified before deployment.

Before any platform touches the water, it must pass a strict off-water component test plan. This phase is all about isolating individual hardware subsystems. It verifies that sensors are reading accurately, actuators are responding sharply, and baseline control loops operate exactly as intended. Catching these component-level quirks in the lab guarantees that the platforms are ready for field operation before introducing environmental uncertainties.



Fig. 10. USV on-water testing at InfinITS Park.

Once cleared, the platforms move to InfinITS Park for real-world field experimentation. Out on the campus lake, the systems face the unpredictable realities of shifting sunlight, wind gusts, water currents, and wireless network interference. Testing scales progressively: first, each platform executes its tasks independently to ensure its specific perception, navigation, and mission logic hold up under environmental stress.

Validating three distinct platforms while simultaneously managing active design cycles presents a significant project management challenge. To accommodate the development schedule, field testing was conducted progressively, with the initial focus placed on independent on-water validation of each platform. With individual platform testing completed, the testing schedule will transition into a dedicated integration phase during the final month of development, when all three platforms will be deployed simultaneously to validate the SoS architecture in real-world conditions. This phase will focus on real-time telemetry exchange and cross-platform mission coordination, allowing the team to identify integration issues such as wireless communication dropouts and timing bottlenecks caused by unexpected delays. The results will be used to refine coordination between platforms and ensure reliable multi-platform operation. A comprehensive breakdown of the test plan protocols and current on-water performance results is provided in Appendix A.

ACKNOWLEDGMENTS

We would like to acknowledge the following institutions and partners for their support:

Dr. Rudy Dikairono, S.T., M.T., M.Sc., Muhtadin, S.T., M.T., and Lukman Hakim, S.T., M.T., Aldinata Rizky Revanda, S.Kom., M.Kom., Yurid Eka Nugraha, S.T., M.Eng. Ph.D., Dr. Mochammad Sahal, S.T., M.Sc., Astria Nur Irfansyah, S.T., M.Eng., Ph.D., for their guidance and advice to the team.

Nur Syahroni, Daniar Fahmi, Mila, Herlis, Maya Sari, Adifan, and Hafidz, Directorate of Student Affairs ITS, for their assistance with administrative requirements and official documentation related to competition funding release.

Junita Dwianingdijah, IKOMA (Ikatan Orang Tua Mahasiswa ITS), for her support in providing financial assistance for the competition.

REFERENCES

[1] RoboNation, “Section 3: Autonomy Challenge,” RobotX Team Handbook, 2026.

[2] OpenCV, “Perspective-n-Point (PnP) Pose Computation,” OpenCV Documentation, n.d. [Online]. Available: https://docs.opencv.org/doc/doxygen/html/d5/d1f/calib3d_solvePnP.html.

[3] Open Navigation LLC, “Smac Hybrid-A* Planner,” Navigation 2 Documentation, n.d. [Online]. Available: https://docs.nav2.org/rolling/configuration_and_development/configuration_guide/planners_plugins/smac/smac_hybrid/configuring_smac_hybrid.html.

[4] BehaviorTree.CPP, “Behavior Trees,” BehaviorTree.CPP Documentation, n.d. [Online]. Available: https://behaviortree.dev/docs/learn-the-basics/BT_basics/.
[5] PX4, “ROS 2 Offboard Control,” PX4 User Guide, n.d. [Online]. Available: https://docs.px4.io/main/en/ros/ros2_offboard_control.

Appendix A: Test Plan & Result

TABLE I. TEST PLAN AND RESULT

Date	Objectives	Results	Location
May 2026 – June 2026	**USV** Virtual system validation using Gazebo simulation Validation of perception, localization, navigation, control, and mission logic **UUV** Simulation-based validation of underwater perception and mission behavior Validation of navigation, pipeline following, pinger-based transit, and mission-state transitions **UAV** Virtual system validation using simulation Validation of flight control, perception, mission state machine, and payload operation	**USV** Navigation and mission logic validated in simulation LiDAR, camera, localization, and obstacle avoidance modules operated under simulated conditions Hybrid A* path planning and control behavior verified across representative mission scenarios **UUV** Simulation confirmed the basic mission sequence and Behavior Tree transitions Pinger-based navigation, pipeline following, repair, and resurfacing behaviors were validated in the simulated environment **UAV** Takeoff, transit, survey, approach, visual positioning, payload operation, return, and landing sequences validated in simulation Object detection and visual positioning pipelines tested under representative scenarios	ITS Robotics Laboratory
June 2026 – July 2026	**USV** Off-water subsystem and hardware validation Sensor, actuator, communication, and mission-payload testing- Integration testing prior to on-water deployment **UAV** Off-water hardware and subsystem validation Camera, flight controller, payload, communication, and mission-control testing Integration testing prior to on-water deployment	**USV** Sensors and actuators operated reliably during ground testing Communication between onboard computers, flight controller, and mission hardware was verified Perception and control modules were integrated and prepared for on-water testing **UAV** Flight-control, camera, communication, and payload interfaces were verified Mission-control software operated correctly with the physical hardware System was prepared for outdoor and on-water Proof of Readiness testing	ITS Robotics Laboratory
July 2026 – August 2026	**USV** On-water Proof of Readiness (POR) testing Mission-level validation under real environmental conditions Validation of perception, localization, navigation, control, communication, and mission execution- Recording of test runs and evidence for POR submission **UUV** No physical deployment; simulation-based validation continued **UAV** On-water Proof of Readiness (POR) testing Validation of flight operation, perception, navigation, communication, and mission execution under real environmental conditions Recording of test runs and evidence for POR submissions	**USV** Vehicle operated successfully on water under real environmental conditions Navigation, perception, control, and mission functions were validated through progressive mission-level testing Communication and vehicle stability were verified during on-water operation Required POR test runs were recorded and documented for submission **UUV** Mission behavior and software functions continued to be validated through simulation Simulation results were used to verify mission logic and identify issues prior to physical deployment **UAV** Flight and mission functions were validated during on-water testing Perception, positioning, communication, and payload operation were evaluated under real operating conditions POR test runs were recorded and documented for submission	InfinITS Park, ITS
August 2026	**USV & UAV** Final Proof of Readiness validation Repeat required mission scenarios and collect final evidence Verify system stability and repeatability under competition-representative conditions Complete documentation and video evidence for submission	**USV & UAV** Required POR scenarios were completed and recorded for submission Final testing confirmed the readiness of the validated systems for competition deployment Test results and supporting video evidence were compiled for POR documentation Remaining software and mission parameters were finalized based on the test results	InfinITS Park, ITS

Appendix B: Risk Management

TABLE I. RISK MANAGEMENT

No.	Risk Description	Root Cause	Potential Impact	Impact Level	Likelihood	Mitigation Strategy	Responsible Team
1	USV system overheats	High CPU/GPU load, continuous high-throttle operation, and limited cooling airflow	Degradation or failure of Mini PC, ESCs, or other onboard electronics	High	Medium	a. Monitor onboard temperature during testing b. Limit prolonged high-throttle operation c. Improve ventilation and heat dissipation	Electrical & Programming
2	Water ingress into USV	Collision with floating obstacles, damaged seals, or shaft/hull tolerance issues	Loss of buoyancy, electrical damage, and potential vehicle failure	High	Low	a. Conduct pre- and post-test leak inspections b. Apply waterproof sealing and reinforcement c. Perform controlled leak testing before deployment	Mechanical & Electrical
3	Loss of communication with GCS	Radio interference, antenna misalignment, network congestion, or connector degradation	Loss of telemetry and inability to monitor or intervene during mission execution	High	Medium	a. Perform pre-deployment communication testing b. Secure antenna placement c. Implement communication-loss handling and vehicle failsafe behavior	Electrical & Programming
4	Inter-vehicle communication failure	Network instability, incompatible interfaces, message loss, or incorrect configuration	Mission synchronization failure and incorrect task execution	High	Medium	a. Define common ROS 2 interfaces and message semantics b. Test communication between vehicles and GCS independently c. Implement message timeout and acknowledgement mechanisms	Programming
5	Insufficient power endurance	Battery degradation, unexpected current spikes, or inaccurate power estimation	Premature mission termination or emergency recovery	High	Medium	a. Select batteries with sufficient safety margin b. Monitor voltage and current in real time c. Validate endurance through representative mission tests	Electrical
6	USV perception failure	Lighting variation, water reflections, sensor noise, occlusion, or unsuitable	Object misclassification, collision risk, or mission failure	High	High	a. Test under varied lighting and environmental conditions b. Augment perception datasets	Programming

		detection parameters				c. Validate camera-LiDAR calibration and fusion before deployment	
7	UAV perception failure	Lighting variation, target appearance, camera motion, altitude variation, or calibration error	Incorrect target localization or failure to complete aerial tasks	High	Medium	a. Validate detection across different altitudes and lighting conditions b. Improve image preprocessing and detection parameters c. Validate camera calibration and georeferencing	Programming
8	UUV localization failure	IMU drift, acoustic measurement noise, depth uncertainty, or underwater visibility limitations	Incorrect navigation and failure to reach the assigned task area	High	Medium	a. Validate localization in simulation b. Tune acoustic and inertial estimation parameters c. Test localization under representative underwater conditions	Programming
9	Navigation drift or unstable control	Sensor noise, electromagnetic interference, inaccurate localization, or poor controller tuning	Path deviation, obstacle collision, or mission failure	High	High	a. Validate navigation and control in simulation b. Tune parameters through progressive physical testing c. Monitor trajectory and control performance during on-water trials	Electrical & Programming
10	Thruster or propulsion failure	Debris entanglement, overheating, mechanical damage, or ESC failure	Reduced maneuverability and inability to maintain the planned trajectory	High	Medium	a. Install protective guards where applicable b. Inspect and clean propulsion units after testing c. Prepare spare propulsion components	Mechanical & Electrical
11	UAV flight-control failure	Incorrect controller parameters, sensor failure, communication loss, or actuator malfunction	Mission interruption, loss of target data, or vehicle recovery requirement	High	Low	a. Validate flight control in simulation b. Perform progressive flight testing c. Configure appropriate flight-controller failsafes	Electrical & Programming
12	UUV actuator failure	Thruster failure, actuator degradation, leakage, or unstable control parameters	Inability to follow the assigned trajectory or complete underwater tasks	High	Medium	a. Test actuators before deployment b. Validate depth and velocity control in simulation c. Prepare spare components and recovery procedures	Mechanical & Electrical

13	Mission mechanism malfunction	Structural vibration, mechanical misalignment, actuator failure, or timing errors	Failure to complete the corresponding mission task	High	Low	a. Reinforce mechanical mounts b. Perform repeated subsystem testing c. Validate mechanism timing under representative operating conditions	Mechanical & Programming
14	Mission synchronization failure	Incorrect task-state handling, delayed communication, or inconsistent mission states	Duplicate tasks, missed tasks, or inefficient mission execution	High	Medium	a. Define explicit mission states and acknowledgements b. Test mission transitions in simulation c. Implement timeout and recovery states	Programming
15	Task queue prioritization failure	Incorrect task priority logic or incomplete mission-state information	Delayed execution of required tasks and reduced mission efficiency	Medium	Medium	a. Define task priority rules before deployment b. Validate queue behavior using simulated mission sequences c. Log task assignment and completion states	Programming
16	GNSS degradation or loss	Signal obstruction, multipath effects, interference, or environmental conditions	Navigation error and deviation from planned mission paths	High	Medium	a. Monitor GNSS quality indicators b. Validate navigation behavior during degraded GNSS conditions c. Provide appropriate fallback behavior	Electrical & Programming
17	Environmental conditions affect testing	Uncontrolled environmental conditions during testing	Reduced system performance, invalid test results, or hardware exposure	Medium	Medium	a. Schedule tests according to weather conditions b. Record environmental conditions during testing c. Prepare waterproofing and test contingency procedures	General
18	Structural damage during operation	Prolonged operation, collisions, inadequate reinforcement, or component fatigue	Misalignment, reduced vehicle performance, or mechanical failure	Medium	Low	a. Conduct periodic structural inspections b. Reinforce critical joints and mounting points c. Inspect vehicles after major test runs	Mechanical
19	Sensor calibration error	Incorrect calibration procedure, sensor mounting changes, or mechanical movement	Incorrect perception, localization, or control behavior	High	Medium	a. Perform calibration before deployment b. Verify sensor mounting and transforms c. Maintain calibration records and repeat calibration after hardware changes	Programming & Electrical

20	Hardware-software interface failure	Driver errors, incorrect configuration, wiring problems, or firmware incompatibility	Loss of sensor or actuator functionality and mission interruption	High	Medium	a. Validate each hardware interface independently b. Maintain tested software and firmware versions c. Perform subsystem testing before full integration	Electrical & Programming
21	Manual recovery failure	Depleted controller battery, damaged communication equipment, or inadequate recovery preparation	Vehicle stranded or prolonged recovery operation	High	Low	a. Conduct pre-deployment controller and recovery equipment checks b. Prepare spare batteries and communication equipment c. Establish a manual recovery procedure before testing	Electrical & Mechanical
22	Loss of vehicle during testing	Severe communication failure, propulsion failure, navigation error, or adverse environmental conditions	Hardware loss and interruption of testing	High	Low	a. Define operating boundaries before testing b. Establish emergency recovery procedures c. Maintain manual intervention and recovery equipment	General
23	Competition configuration mismatch	Last-minute software, hardware, parameter, or network changes	Previously validated behavior may not be reproduced during competition	High	Medium	a. Freeze the validated configuration before deployment b. Maintain version-controlled software and parameter files c. Perform final regression testing after configuration changes	Programming & Electrical
24	Unsafe multi-vehicle operation	Inadequate operating boundaries, unexpected vehicle behavior, or poor mission coordination	Collision, vehicle damage, or interruption of multiple missions	High	Low	a. Define operational boundaries and separation rules b. Test multi-vehicle operation progressively c. Establish emergency stop and recovery procedures	General & Programming